AMENDMENTS TO THE CLAIMS

Please amend claims 28-35 as follows:

1. (Original) A method for managing a cache, the method comprising:

assigning a cache priority to each of a plurality of accessed item as a function of the item's size, retrieval cost and access frequency;

dynamically updating cache priorities as items are accessed; and

determining which items to store in the cache as a function of cache priority.

2. (Original) The method of claim 1 further comprising:

calculating an item's size relative to the size of the cache.

3. (Original) The method of claim 2 wherein calculating an item's size relative to the size of the cache further comprises:

dividing the size of the item by the size of the cache.

4. (Original) The method of claim 1 further comprising:

calculating an item's retrieval cost as a function of the item's retrieval time and the item's size.

5. (Original) The method of claim 4 wherein calculating an item's retrieval cost as a function of the item's retrieval time and the item's size:

dividing the retrieval time of the item by the size of the item.

6. (Original) The method of claim 1 further comprising:

calculating an item's access frequency relative to access frequency for other items.

7. (Original) The method of claim 6 wherein calculating an item's access frequency relative to access frequency for other items further comprises:

dividing a number of requests for the item during a period of time by a total number of requests for items during the period of time.

8. (Original) The method of claim 1 further comprising:

calculating cache priority for an item by multiplying the item's size, retrieval cost and access frequency.

9. (Original) The method of claim 1 further comprising:

each time access to an item is requested, determining whether the requested item has been assigned a cache priority; and

performing a step from a group of steps consisting of:

responsive to determining that the requested item has not been assigned a cache priority, calculating a cache priority and assigning the calculated cache priority to the requested item; and

responsive to determining that the requested item has been assigned a cache priority, updating the cache priority to reflect the request for the item.

10. (Original) The method of claim 1 further comprising:

maintaining a sorted list of associations between each accessed item and its cache priority;

each time access to an item is requested, determining whether the requested item has been assigned a cache priority by reading the sorted list; and

performing a step from a group of steps consisting of:

responsive to determining that the requested item has not been assigned a cache priority, calculating a cache priority and adding an entry associating the requested item with the cache priority to the sorted list; and

responsive to determining that the requested item has been assigned a cache priority, updating the requested item's entry in the sorted list to reflect the request for the item.

11. (Original) The method of claim 1 wherein determining which items to store in the cache as a function of cache priority further comprises:

receiving a request for an item not in the cache;

retrieving the item;

determining that the cache is full;

comparing the cache priority of the retrieved item to the cache priority of each item in the cache; and

performing a step from a group of steps consisting of:

responsive to determining that the cache priority of at least one item in the cache is lower than the cache priority of the retrieved item, overwriting a cached item with the lowest cache priority with the retrieved item; and

responsive to determining that no item in the cache has a cache priority lower than the retrieved item, not storing the retrieved item in the cache.

12. (Original) A computer readable medium containing a computer program product for managing a cache, the computer readable medium comprising:

program code for assigning a cache priority to each of a plurality of accessed item as a function of the item's size, retrieval cost and access frequency;

program code for dynamically updating cache priorities as items are accessed; and

program code for determining which items to store in the cache as a function of cache priority.

13. (Original) The computer program product of claim 12 further comprising:

program code for calculating an item's size relative to the size of the cache by dividing the size of the item by the size of the cache.

14. (Original) The computer program product of claim 12 further comprising:

program code for calculating an item's retrieval cost as a function of the item's retrieval time and the item's size by dividing the retrieval time of the item by the size of the item.

15. (Original) The computer program product of claim 12 further comprising:

program code for calculating an item's access frequency relative to access frequency for other items by dividing a number of requests for the item during a period of time by a total number of requests for items during the period of time.

16. (Original) The computer program product of claim 12 further comprising:

program code for calculating cache priority for an item by multiplying the item's size, retrieval cost and access frequency.

17. (Original) The computer program product of claim 12 further comprising:

program code for, each time access to an item is requested, determining whether the requested item has been assigned a cache priority; and

program code for performing a step from a group of steps consisting of:

responsive to determining that the requested item has not been assigned a cache priority, calculating a cache priority and assigning the calculated cache priority to the requested item; and

responsive to determining that the requested item has been assigned a cache priority, updating the cache priority to reflect the request for the item.

18. (Original) The computer program product of claim 12 further comprising:

program code for maintaining a sorted list of associations between each accessed item and its cache priority;

program code for, each time access to an item is requested, determining whether the requested item has been assigned a cache priority by reading the sorted list; and

program code for performing a step from a group of steps consisting of:

responsive to determining that the requested item has not been assigned a cache priority, calculating a cache priority and adding an entry associating the requested item with the cache priority to the sorted list; and

responsive to determining that the requested item has been assigned a cache priority, updating the requested item's entry in the sorted list to reflect the request for the item.

19. (Original) The computer program product of claim 12 wherein the program code for determining which items to store in the cache as a function of cache priority further comprises:

program code for receiving a request for an item not in the cache;

program code for retrieving the item;

program code for determining that the cache is full;

program code for comparing the cache priority of the retrieved item to the cache priority of each item in the cache; and

program code for performing a step from a group of steps consisting of:

responsive to determining that the cache priority of at least one item in the cache is lower than the cache priority of the retrieved item, overwriting a cached item with the lowest cache priority with the retrieved item; and

responsive to determining that no item in the cache has a cache priority lower than the retrieved item, not storing the retrieved item in the cache.

20. (Original) A computer system for managing a cache, the computer system comprising:

means for assigning a cache priority to each of a plurality of accessed item as a function of the item's size, retrieval cost and access frequency;

means for dynamically updating cache priorities as items are accessed; and

means for determining which items to store in the cache as a function of cache priority.

21. (Original) The computer system of claim 20 further comprising:

means for calculating an item's size relative to the size of the cache by dividing the size of the item by the size of the cache.

22. (Original) The computer system of claim 20 further comprising:

means for calculating an item's retrieval cost as a function of the item's retrieval time and the item's size by dividing the retrieval time of the item by the size of the item.

23. (Original) The computer system of claim 20 further comprising:

means for calculating an item's access frequency relative to access frequency for other items by dividing a number of requests for the item during a period of time by a total number of requests for items during the period of time.

24. (Original) The computer system of claim 20 further comprising:

means for calculating cache priority for an item by multiplying the item's size, retrieval cost and access frequency.

25. (Original) The computer system of claim 20 further comprising:

means for, each time access to an item is requested, determining whether the requested item has been assigned a cache priority; and

means for performing a step from a group of steps consisting of:

responsive to determining that the requested item has not been assigned a cache priority, calculating a cache priority and assigning the calculated cache priority to the requested item; and

responsive to determining that the requested item has been assigned a cache priority, updating the cache priority to reflect the request for the item.

26. (Original) The computer system of claim 20 further comprising:

means for maintaining a sorted list of associations between each accessed item and its cache priority;

means for, each time access to an item is requested, determining whether the requested item has been assigned a cache priority by reading the sorted list; and

means for performing a step from a group of steps consisting of:

responsive to determining that the requested item has not been assigned a cache priority, calculating a cache priority and adding an entry associating the requested item with the cache priority to the sorted list; and

responsive to determining that the requested item has been assigned a cache priority, updating the requested item's entry in the sorted list to reflect the request for the item.

27. (Original) The computer system of claim 20 wherein the means for determining which items to store in the cache as a function of cache priority further comprises:

means for receiving a request for an item not in the cache;

means for retrieving the item;

means for determining that the cache is full;

means for comparing the cache priority of the retrieved item to the cache priority of each item in the cache; and

means for performing a step from a group of steps consisting of:

responsive to determining that the cache priority of at least one item in the cache is lower than the cache priority of the retrieved item, overwriting a cached item with the lowest cache priority with the retrieved item; and

responsive to determining that no item in the cache has a cache priority lower than the retrieved item, not storing the retrieved item in the cache.

28. (Currently Amended) A computer system for managing a cache, the computer system comprising:

a ~~software portion~~ priority assigner configured to assign a cache priority to each of a plurality of accessed item as a function of the item's size, retrieval cost and access frequency;

a ~~software portion~~ priority updater configured to dynamically update cache priorities as items are accessed; and

a ~~software portion~~ storage determiner configured to determine which items to store in the cache as a function of cache priority.

29. (Currently Amended) The computer system of claim 28 further comprising:

a ~~software portion~~ size calculater configured to calculate an item's size relative to the size of the cache by dividing the size of the item by the size of the cache.

30. (Currently Amended) The computer system of claim 28 further comprising:

a ~~software portion~~ cost calculater configured to calculate an item's retrieval cost as a function of the item's retrieval time and the item's size by dividing the retrieval time of the item by the size of the item.

31. (Currently Amended) The computer system of claim 28 further comprising:

a ~~software portion~~ frequency calculater configured to calculate an item's access frequency relative to access frequency for other items by dividing a number of requests for the item during a period of time by a total number of requests for items during the period of time.

32. (Currently Amended) The computer system of claim 28 further comprising:

a ~~software portion~~ priority calculater configured to calculate cache priority for an item by multiplying the item's size, retrieval cost and access frequency.

33. (Currently Amended) The computer system of claim 28 further comprising:

~~a software portion~~ an existence determiner configured to determine, each time access to an item is requested, whether the requested item has been assigned a cache priority; and

a ~~software portion~~ priority determiner configured to perform a step from a group of steps consisting of:

responsive to determining that the requested item has not been assigned a cache priority, calculating a cache priority and assigning the calculated cache priority to the requested item; and

responsive to determining that the requested item has been assigned a cache priority, updating the cache priority to reflect the request for the item.

34. (Currently Amended) The computer system of claim 28 further comprising:

~~a software portion~~ an association maintainer configured to maintain a sorted list of associations between each accessed item and its cache priority;

~~a software portion~~ an existence determiner configured to determine, each time access to an item is requested, whether the requested item has been assigned a cache priority by reading the sorted list; and

a ~~software portion~~ priority determiner configured to perform a step from a group of steps consisting of:

responsive to determining that the requested item has not been assigned a cache priority, calculating a cache priority and adding an entry associating the requested item with the cache priority to the sorted list; and

responsive to determining that the requested item has been assigned a cache priority, updating the requested item's entry in the sorted list to reflect the request for the item.

35. (Currently Amended) The computer system of claim 28 wherein the ~~software portion configured to determine which items to store in the cache as a function of cache priority~~ storage determiner further comprises:

a ~~software portion~~ request receiver configured to receive a request for an item not in the cache;

~~a software portion~~ an item retriever configured to retrieve the item;

a ~~software portion~~ fullness determiner configured to determine that the cache is full;

a ~~software portion~~ priority comparer configured to compare the cache priority of the retrieved item to the cache priority of each item in the cache; and

a ~~software portion~~ cache modifier configured to perform a step from a group of steps consisting of:

responsive to determining that the cache priority of at least one item in the cache is lower than the cache priority of the retrieved item, overwriting a cached item with the lowest cache priority with the retrieved item; and

responsive to determining that no item in the cache has a cache priority lower than the retrieved item, not storing the retrieved item in the cache.